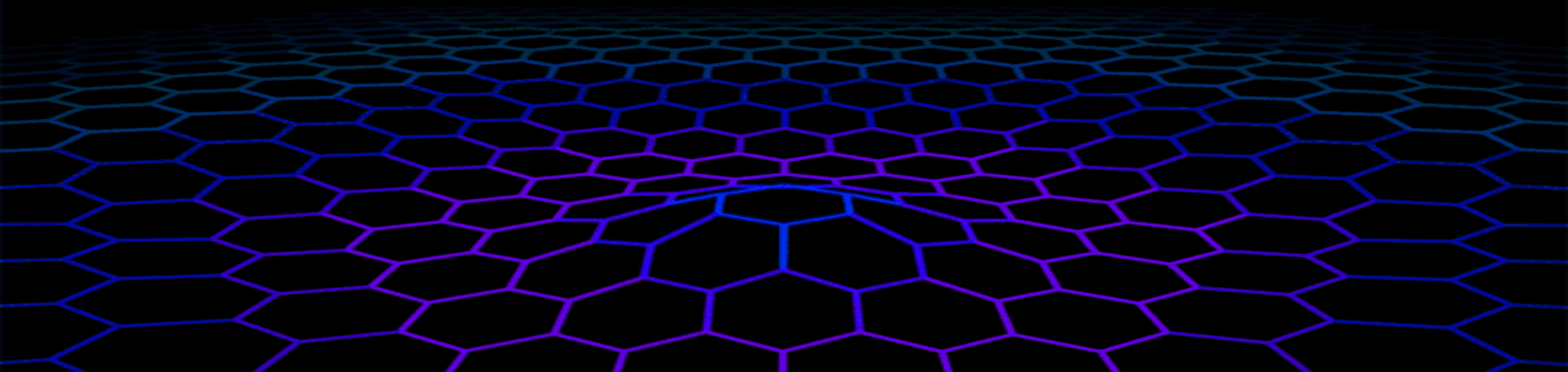
Unisenac
Campus Pelotas

Banco de Dados 1



Resolução dos exercícios - cinema

CINEMINHA

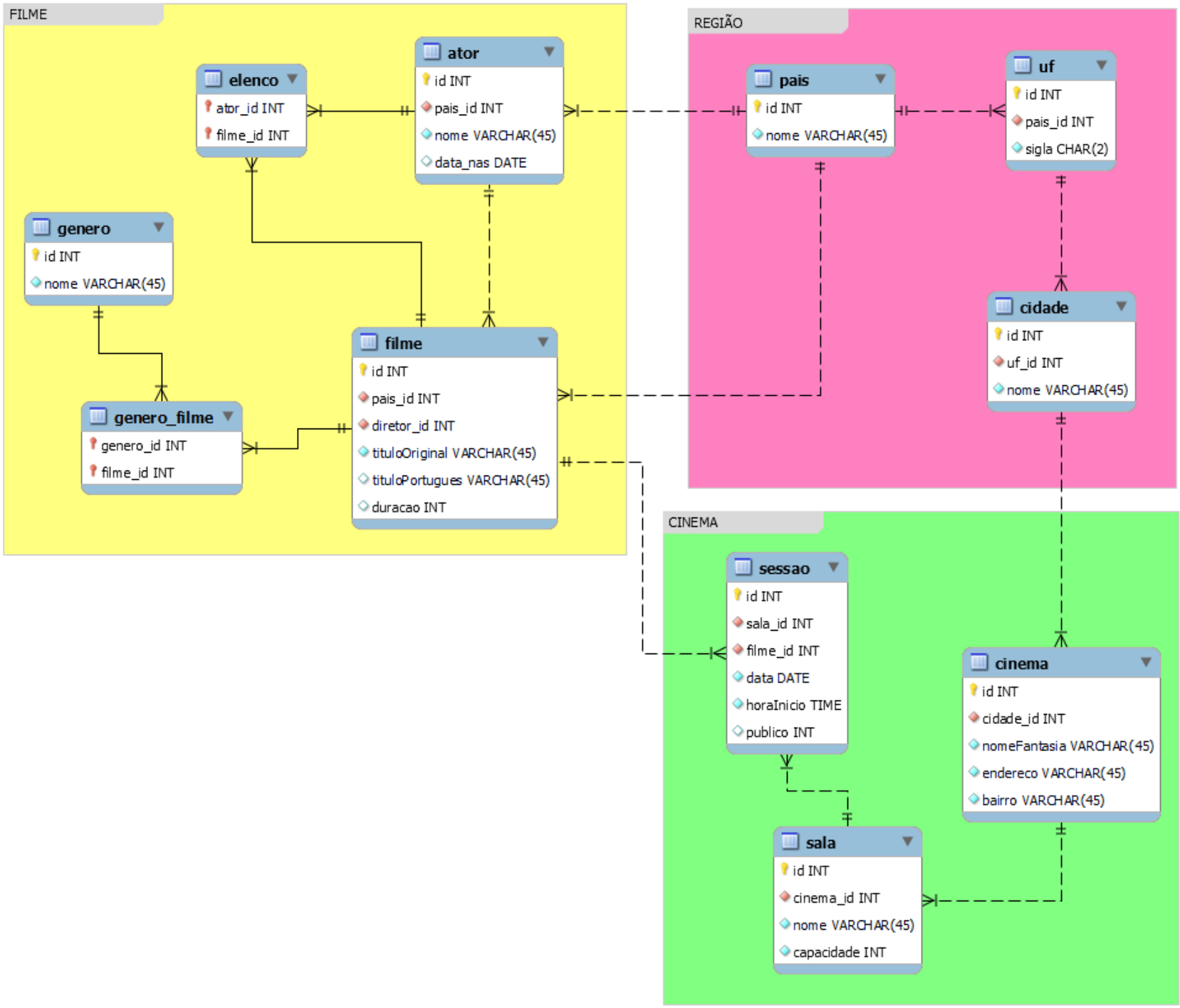
CINEMINHA

O Banco de dados a seguir foi projetado para se adaptar às necessidades de um controle de cinemas e filmes da empresa CINEMINHA.

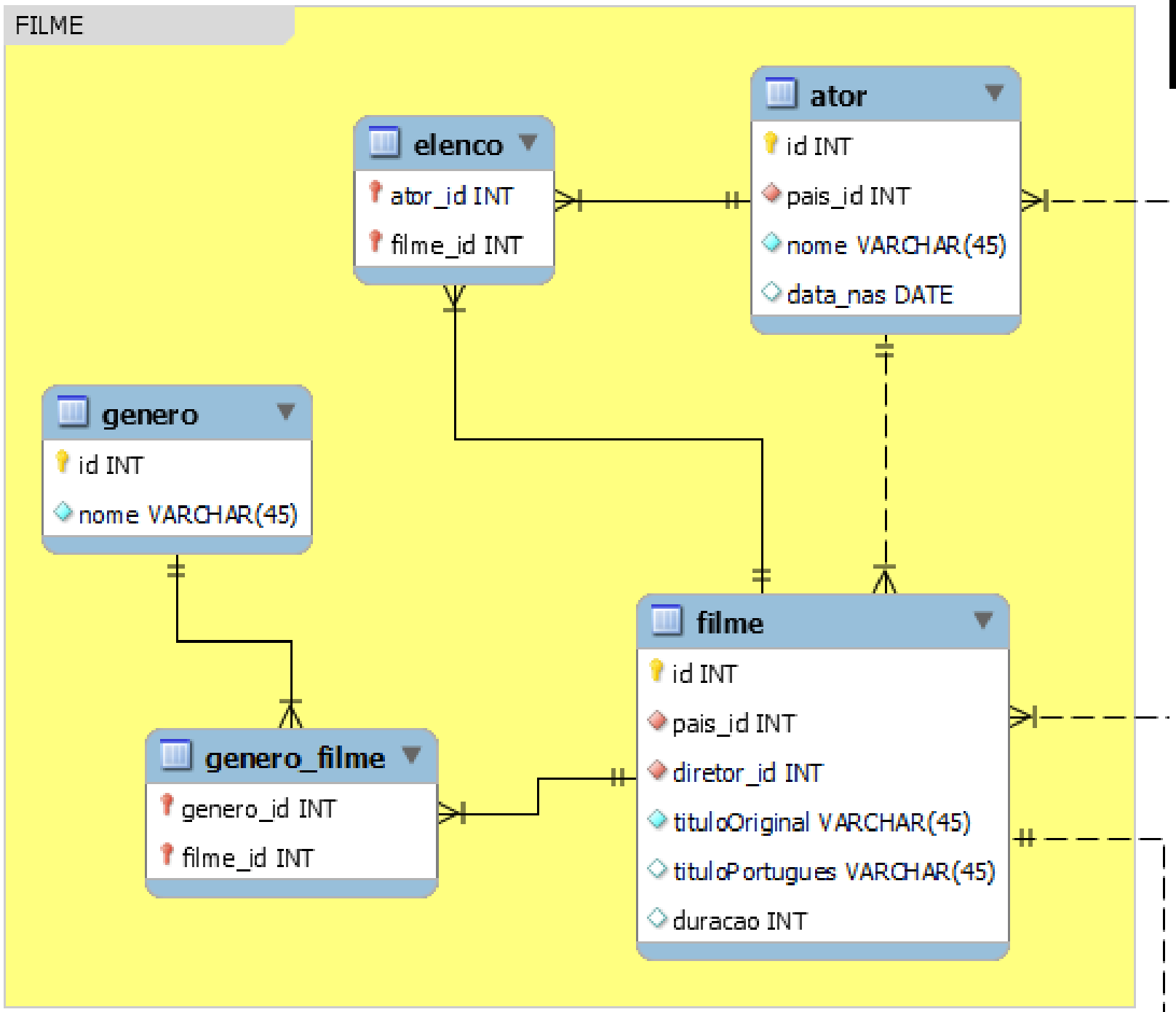
A modelagem, seguiu algumas definições e regras:

1. A empresa de distribuição possui vários cinemas em várias localidades;
2. Cada cinema possui uma identificação única, um nome fantasia e um endereço;
3. Cada cinema possui “n” salas. Cada uma com identificação única e sua capacidade de lotação;
4. Os filmes podem ser dos mais variados tipos e gêneros;
5. Cada filme é registrado com um título original, e se for filme estrangeiro, possuirá também o título em português, o gênero, sua duração, seu país de origem, informações sobre os atores que compõem seu elenco, e o seu diretor. Existe um único diretor para cada filme;
6. Alguns cinemas apresentam mais de um filme em cartaz (em sessões e/ou salas diferentes);
7. As sessões possuem horários determinados;
8. Os atores de um filme podem, obviamente, atuar em diversos filmes, assim como o diretor de um filme pode também ser ator nesse filme, ou ator em outro filme. Um ator possui como característica de identificação: nome, nacionalidade e data de nascimento;
9. As sessões do cinema devem ter seu público registrado diariamente, para que se permita a totalização do público quando o filme sair de cartaz, ou a qualquer instante.

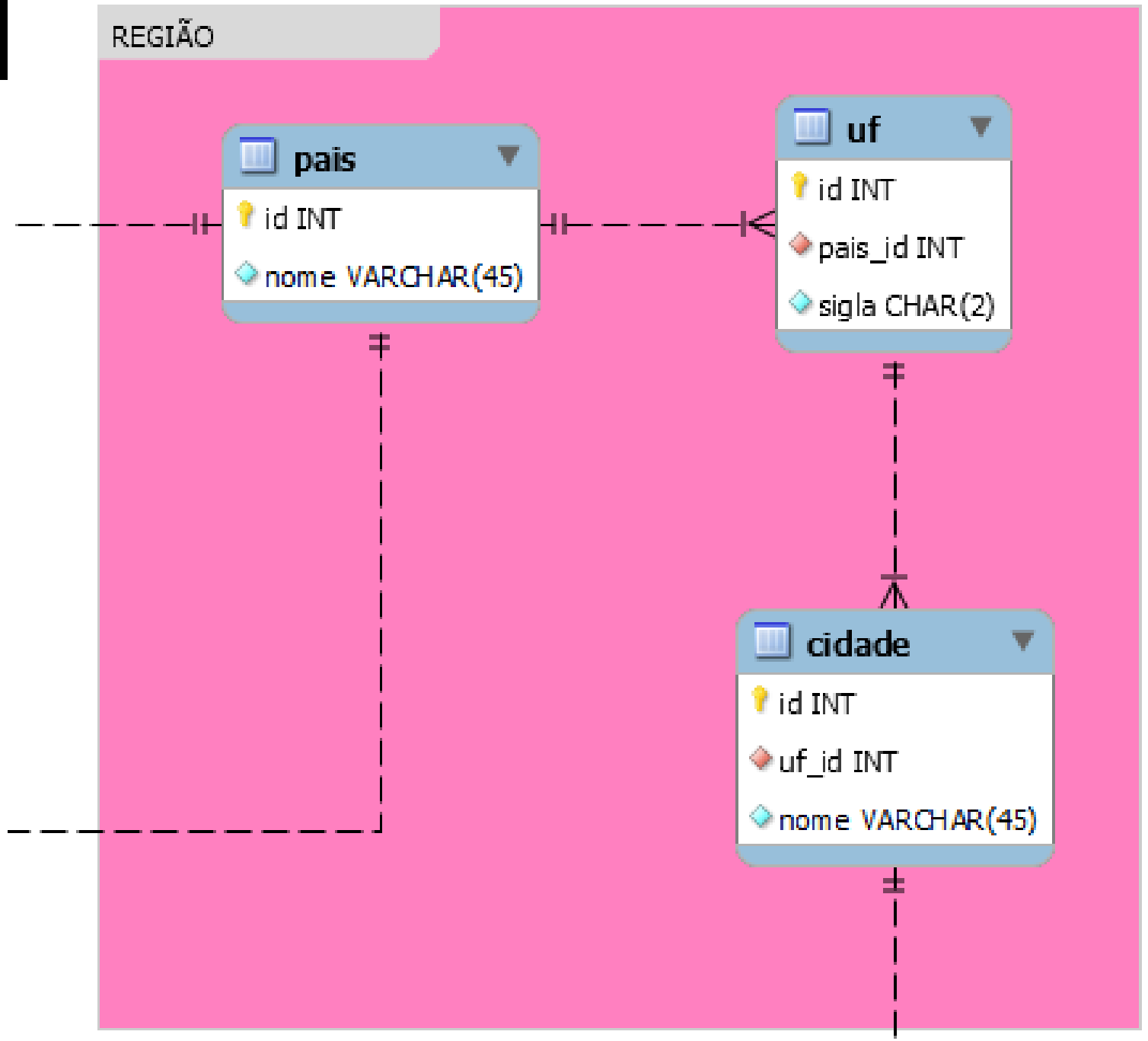
CINEMINHA


FILME
elenco
ator_id INT
filme_id INT
ator
id INT
pais_id INT
nome VARCHAR(45)
data_nas DATE
genero
id INT
nome VARCHAR(45)
genero_filme
genero_id INT
filme_id INT
filme
id INT
pais_id INT
diretor_id INT
tituloOriginal VARCHAR(45)
tituloPortugues VARCHAR(45)
duracao INT
REGIÃO
pais
id INT
nome VARCHAR(45)
uf
id INT
pais_id INT
sigla CHAR(2)
cidade
id INT
uf_id INT
nome VARCHAR(45)
CINEMA
sessao
id INT
sala_id INT
filme_id INT
data DATE
horaInicio TIME
publico INT
cinema
id INT
cidade_id INT
nomeFantasia VARCHAR(45)
endereco VARCHAR(45)
bairro VARCHAR(45)
sala
id INT
cinema_id INT
nome VARCHAR(45)
capacidade INT

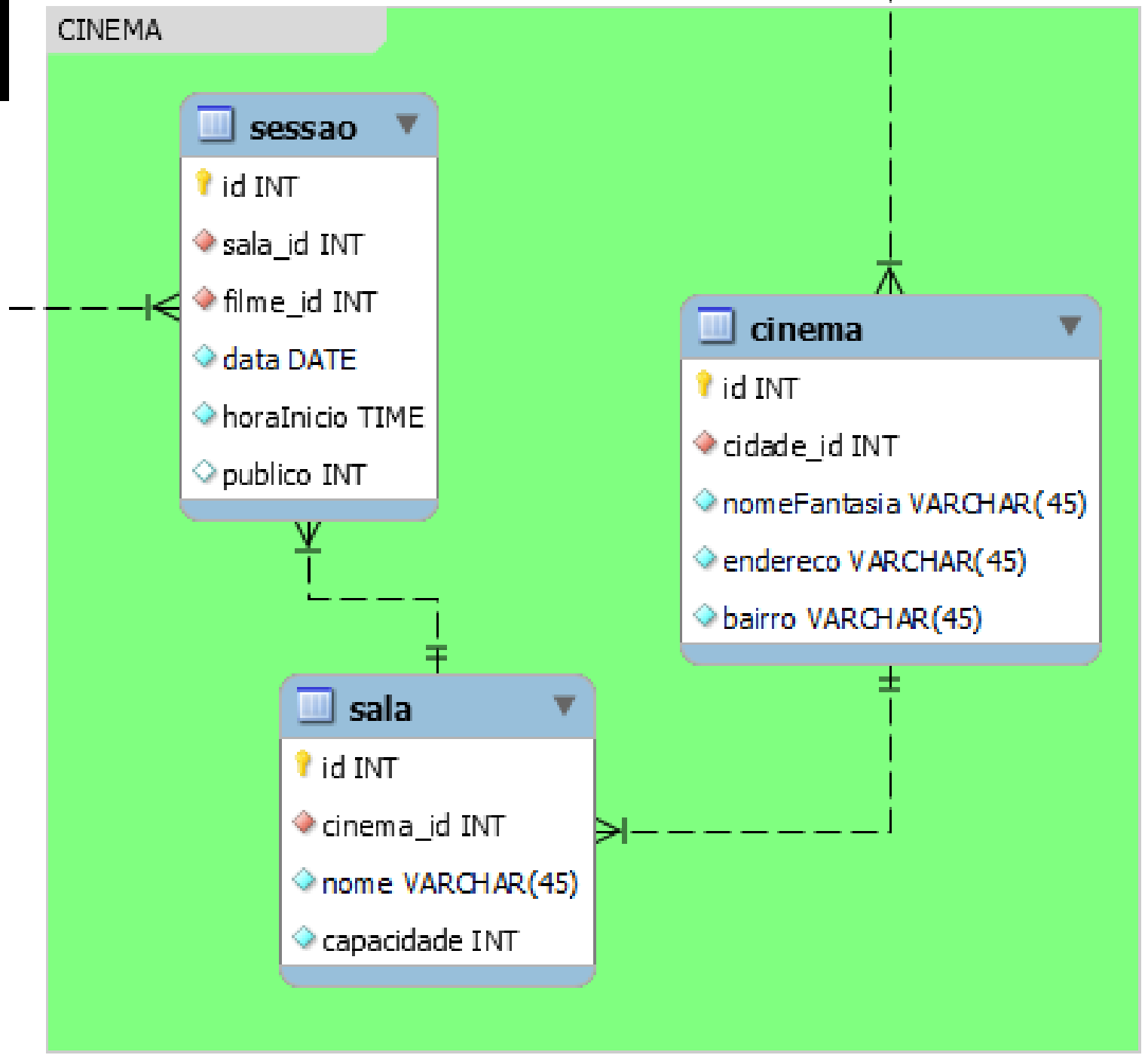
CINEMINHA


FILME
ator
id INT
pais_id INT
nome VARCHAR(45)
data_nas DATE
elenco
ator_id INT
filme_id INT
genero
id INT
nome VARCHAR(45)
filme
id INT
pais_id INT
diretor_id INT
tituloOriginal VARCHAR(45)
tituloPortugues VARCHAR(45)
duracao INT
genero_filme
genero_id INT
filme_id INT

CINEMINHA


REGIÃO
pais
id INT
nome VARCHAR(45)
uf
id INT
pais_id INT
sigla CHAR(2)
cidade
id INT
uf_id INT
nome VARCHAR(45)

CINEMINHA


CINEMA
sessao
id INT
sala_id INT
filme_id INT
data DATE
horaInicio TIME
publico INT
cinema
id INT
cidade_id INT
nomeFantasia VARCHAR(45)
endereco VARCHAR(45)
bairro VARCHAR(45)
sala
id INT
cinema_id INT
nome VARCHAR(45)
capacidade INT

Crie as consultas

a) Apuração do público por município

b) Apuração do público por cinema

c) Apuração do público por sessão de cada cinema;

d) Permitir uma forma de consulta, que dado um determinado ator, sejam localizados todos os cinemas onde estão (ou estiveram) em cartaz os filmes em que este ator atua;

e) Em quais cinemas está sendo exibido um determinado gênero de filme;

f) Em quais cinemas estão sendo exibidos filmes nacionais;

```sql
-- a.  Apuração do público por município
SELECT SUM(sessao.publico) AS Publico,
       cidade.nome         AS Cidade
FROM   sessao
       INNER JOIN sala   ON sessao.sala_id   = sala.id
       INNER JOIN cinema ON sala.cinema_id   = cinema.id
       INNER JOIN cidade ON cinema.cidade_id = cidade.id
-- WHERE sessao.data = "2024-06-12"
GROUP  BY cidade.nome
ORDER  BY cidade.nome;
```

```sql
-- b.  Apuração do público por cinema
SELECT SUM(sessao.publico) AS Qtde,
       cinema.nomefantasia AS Cinema
FROM   sessao
       INNER JOIN sala   ON sessao.sala_id = sala.id
       INNER JOIN cinema ON sala.cinema_id = cinema.id
-- WHERE sessao.data = "2024-06-12"
GROUP  BY cinema.nomefantasia
ORDER  BY cinema.nomefantasia;
```

```sql
-- c.  Apuração do público por sessão de cada cinema
SELECT cinema.nomefantasia AS Cinema,
       sessao.horainicio   AS Horario,
       SUM(sessao.publico) AS Qtde
FROM   sessao
       INNER JOIN sala   ON sessao.sala_id = sala.id
       INNER JOIN cinema ON cinema.id      = sala.cinema_id
GROUP  BY sessao.horainicio,
          cinema.nomefantasia
ORDER  BY cinema.nomefantasia,
          sessao.horainicio;
```

```sql
/* d.  Dado um determinado ator, sejam localizados todos os cinemas
 onde estão em cartaz os filmes em que este ator atua */
SELECT cinema.nomefantasia  AS Cinema,
       filme.titulooriginal AS Filme,
       ator.nome            AS Ator
FROM   cinema
       INNER JOIN sala   ON cinema.id = sala.cinema_id
       INNER JOIN sessao ON sala.id   = sessao.sala_id
       INNER JOIN filme  ON filme.id  = sessao.filme_id
       INNER JOIN elenco ON filme.id  = elenco.filme_id
       INNER JOIN ator   ON ator.id   = elenco.ator_id
WHERE  ator.nome = 'Jason Clarke'
--     AND sessao.data = CURDATE();
       AND sessao.data = "2024-06-12";
```

```sql
-- e.  Em quais cinemas está sendo exibido um determinado gênero de
filme
SELECT DISTINCT cidade.nome          AS Cidade,
                cinema.nomefantasia  AS Cinema,
                filme.titulooriginal AS Filme,
                genero.nome          AS Genero
FROM   cidade
       INNER JOIN cinema       ON cidade.id = cinema.cidade_id
       INNER JOIN sala         ON cinema.id = sala.cinema_id
       INNER JOIN sessao       ON sala.id   = sessao.sala_id
       INNER JOIN filme        ON filme.id  = sessao.filme_id
       INNER JOIN genero_filme ON filme.id  = genero_filme.filme_id
       INNER JOIN genero       ON genero.id = genero_filme.genero_id
WHERE  genero.nome = 'Drama'
--     AND sessao.data = CURDATE();
       AND sessao.data = "2024-06-12";
```

```sql
-- f.  Em quais cinemas estão sendo exibidos filmes nacionais
SELECT DISTINCT cidade.nome          AS Cidade,
                cinema.nomefantasia  AS Cinema,
                filme.titulooriginal AS Filme
FROM   cidade
       INNER JOIN cinema ON cidade.id = cinema.cidade_id
       INNER JOIN sala   ON cinema.id = sala.cinema_id
       INNER JOIN sessao ON sala.id   = sessao.sala_id
       INNER JOIN filme  ON filme.id  = sessao.filme_id
       INNER JOIN pais   ON pais.id   = filme.pais_id
WHERE  pais.nome = 'Brasil'
--     AND sessao.data = CURDATE();
       AND sessao.data = "2024-06-03";
```

+ SQL

SubConsultas / Consultas Aninhadas

```sql
/* Pode ser implementada com o uso de blocos SELECT-FROM-WHERE dentro
da cláusula WHERE de outra consulta.
Pode ser implementada com o operador IN, que compara um valor com um
conjunto de valores ao mesmo tempo.
Exemplo 1 */

SELECT cinema.nomeFantasia AS Cinema,
       cidade.nome         AS Cidade
FROM cinema
INNER JOIN cidade ON cinema.cidade_id = cidade.id
WHERE cidade.nome IN ('Pelotas', 'Porto Alegre')
ORDER BY cidade.nome;
```

SubConsultas / Consultas Aninhadas

```sql
/* Pode ser implementada com o uso de blocos SELECT-FROM-WHERE dentro
da cláusula WHERE de outra consulta.
Pode ser implementada com o operador IN, que compara um valor com um
conjunto de valores ao mesmo tempo.
Exemplo 2 */

SELECT cinema.nomeFantasia AS Cinema,
       cidade.nome         AS Cidade
FROM cinema
INNER JOIN cidade ON cinema.cidade_id = cidade.id
WHERE cidade.nome IN (SELECT nome FROM cidade WHERE cidade.uf_id = 1)
ORDER BY cidade.nome;
```

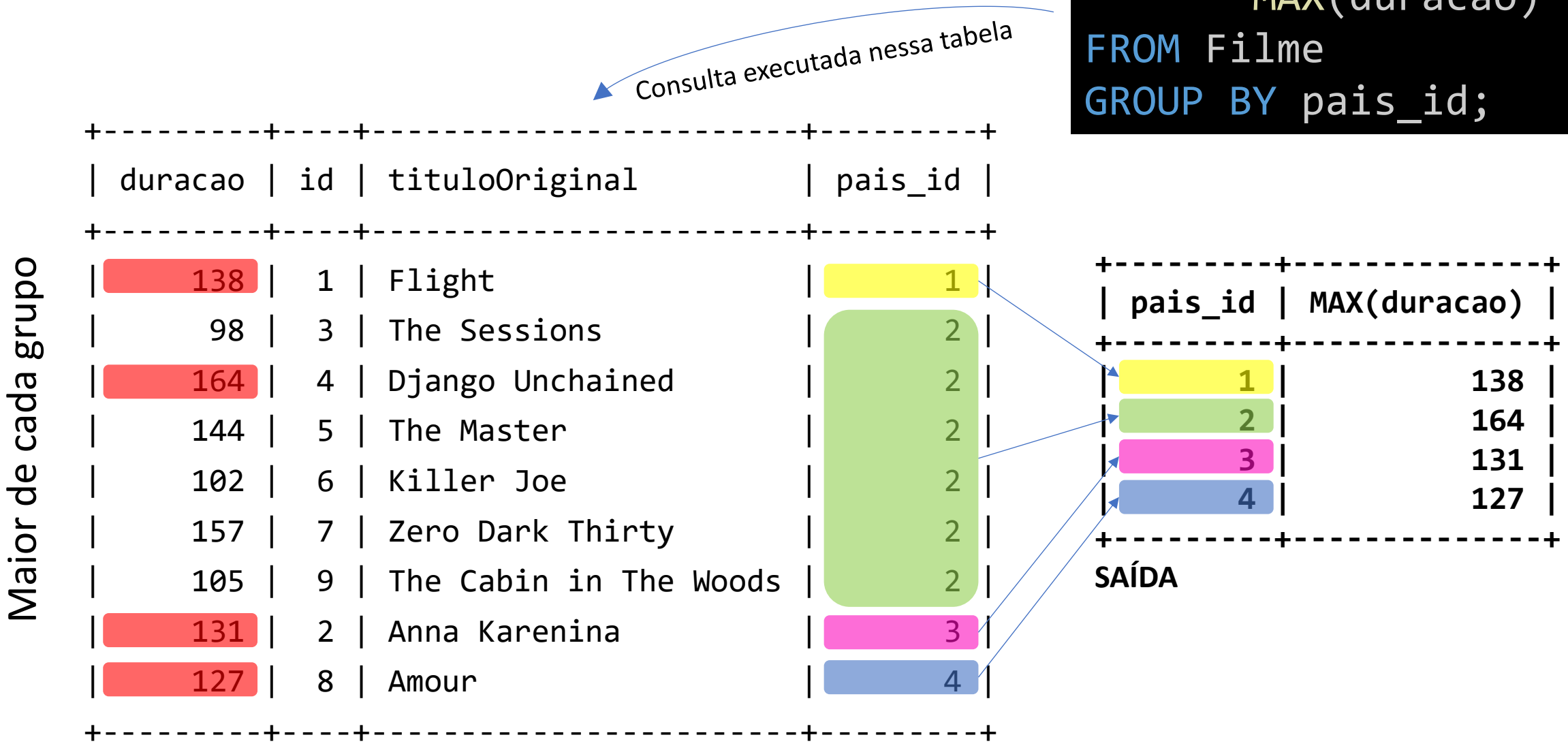
Verificando com a estrutura de uma tabela foi criada

```
SHOW CREATE TABLE ator;
+-------+-----------------------------------------------------+
| Table | Create Table
+-------+-----------------------------------------------------+
| ator  | CREATE TABLE ator (
  id          int NOT NULL AUTO_INCREMENT,
  pais_id     int NOT NULL,
  nome        varchar(45) NOT NULL,
  data_nas    date DEFAULT NULL,
  PRIMARY KEY (id),
  KEY fk_ator_pais (pais_id),
  CONSTRAINT fk_ator_pais FOREIGN KEY (pais_id) REFERENCES pais (id)
) ENGINE=InnoDB AUTO_INCREMENT=39 |
+-------+-----------------------------------------------------+
```

GROUP BY

Agrupa os resultados por uma coluna selecionada

```
SELECT pais_id,
       MAX(duracao)
FROM Filme
GROUP BY pais_id;
```



MIN

```
SELECT MIN(duracao)
FROM filme;
+---------------+
| MIN(duracao) |
+---------------+
|           98 |
+---------------+
1 row in set (0.00 sec)
```

MAX

```
SELECT MAX(duracao)
FROM filme;
+---------------+
| MAX(duracao) |
+---------------+
|          164 |
+---------------+
1 row in set (0.00 sec)
```

MAX

```
SELECT pais_id,
       MAX(duracao)
FROM Filme
GROUP BY pais_id;
+---------+---------------+
| pais_id | MAX(duracao) |
+---------+---------------+
|       1 |           138 |
|       2 |           164 |
|       3 |           131 |
|       4 |           127 |
+---------+---------------+
4 rows in set (0.00 sec)
```

AVG

```
SELECT AVG(duracao)
FROM filme;
+---------------+
| AVG(duracao) |
+---------------+
|     129.5556 |
+---------------+
1 row in set (0.00 sec)
```

SUM

```
SELECT SUM(duracao)
FROM filme;
+---------------+
| SUM(duracao) |
+---------------+
|         1166 |
+---------------+
1 row in set (0.00 sec)
```

LIMIT

```
SELECT * FROM filme LIMIT 5;
+----+---------+------------+------------------+------------------+----------+
| id | pais_id | diretor_id | tituloOriginal   | tituloPortugues | duracao |
+----+---------+------------+------------------+------------------+----------+
|  1 |       1 |          3 | Flight           | O Voo            |      138 |
|  2 |       3 |          3 | Anna Karenina    | Anna Karenina    |      131 |
|  3 |       2 |          1 | The Sessions     | As Sessões       |       98 |
|  4 |       2 |          7 | Django Unchained | Django Livre     |      164 |
|  5 |       2 |          3 | The Master       | O Mestre         |      144 |
+----+---------+------------+------------------+------------------+----------+
5 rows in set (0.00 sec)
```

COUNT

```
SELECT COUNT(*)
FROM filme
WHERE pais_id = 2;
+----------+
| COUNT(*) |
+----------+
|        6 |
+----------+
1 row in set (0.00 sec)
```

RIGHT

```
SELECT RIGHT(titulooriginal, 10)
FROM filme;
+---------------------------+
| RIGHT(titulooriginal, 10) |
+---------------------------+
| Flight                    |
| a Karenina                |
| e Sessions                |
|  Unchained                |
| The Master                |
| Killer Joe                |
| ark Thirty                |
| Amour                     |
|  The Woods                |
+---------------------------+
9 rows in set (0.00 sec)
```

LEFT

```
SELECT LEFT(titulooriginal, 10)
FROM filme;
+--------------------------+
| LEFT(titulooriginal, 10) |
+--------------------------+
| Flight                   |
| Anna Karen               |
| The Sessio               |
| Django Unc               |
| The Master               |
| Killer Joe               |
| Zero Dark                |
| Amour                    |
| The Cabin                |
+--------------------------+
9 rows in set (0.00 sec)
```

UPPER / UCASE

```
SELECT UPPER(nome)
FROM genero;
+--------------+
| UPPER(nome) |
+--------------+
| COMÉDIA      |
| FICÇÃO       |
| DRAMA        |
| AÇÃO         |
| SUSPENSE     |
| TERROR       |
| FAROESTE     |
| AVENTURA     |
+--------------+
8 rows in set (0.00 sec)
```

LOWER / LCASE

```
SELECT LOWER(nome)
FROM genero;
+--------------+
| LOWER(nome) |
+--------------+
| comédia      |
| ficção       |
| drama        |
| ação         |
| suspense     |
| terror       |
| faroeste     |
| aventura     |
+--------------+
8 rows in set (0.00 sec)
```

REVERSE

```
SELECT REVERSE(nome)
FROM genero;
+----------------+
| REVERSE(nome) |
+----------------+
| aidémoC        |
| oãçciF         |
| amarD          |
| oãçA           |
| esnepsuS       |
| rorreT         |
| etseoraF       |
| arutnevA       |
+----------------+
8 rows in set (0.00 sec)
```

Tabela USUARIO

```
SELECT nome from usuario;
+-------------------------+
| nome                    |
+-------------------------+
| .Leonardo da Silva      |
| ..Cleber Duarte         |
| Débora Rodrigues        |
| Mario Fernandes         |
| Gustavo Borges..        |
| Valéria Santiago....    |
+-------------------------+
6 rows in set (0.00 sec)
```

LTRIM

```
SELECT LTRIM(nome)
FROM usuario;
+-------------------------+
| LTRIM(nome)             |
+-------------------------+
| Leonardo da Silva       |
| Cleber Duarte           |
| Débora Rodrigues        |
| Mario Fernandes         |
| Gustavo Borges..        |
| Valéria Santiago.....   |
+-------------------------+
6 rows in set (0.00 sec)
```

RTRIM

```
SELECT RTRIM(nome)
    -> FROM usuario;
+----------------------+
| RTRIM(nome)          |
+----------------------+
|  Leonardo da Silva |
|  Cleber Duarte     |
| Débora Rodrigues   |
| Mario Fernandes    |
| Gustavo Borges     |
| Valéria Santiago   |
+----------------------+
6 rows in set (0.00 sec)
```

TRIM

```
SELECT TRIM(nome)
FROM usuario;
+--------------------+
| TRIM(nome)         |
+--------------------+
| Leonardo da Silva |
| Cleber Duarte      |
| Débora Rodrigues   |
| Mario Fernandes    |
| Gustavo Borges     |
| Valéria Santiago   |
+--------------------+
6 rows in set (0.00 sec)
```

Tabela USUARIO

```
SELECT nome from usuario;
+-------------------------+
| nome                    |
+-------------------------+
| .Leonardo da Silva      |
| ..Cleber Duarte         |
| Débora Rodrigues        |
| Mario Fernandes         |
| Gustavo Borges..        |
| Valéria Santiago....    |
+-------------------------+
6 rows in set (0.00 sec)
```

LENGTH

```
SELECT nome, LENGTH(nome)
FROM usuario;
+----------------------+--------------+
| nome                 | LENGTH(nome) |
+----------------------+--------------+
| .Leonardo da Silva   |           18 |
| ..Cleber Duarte      |           15 |
| Débora Rodrigues     |           17 |
| Mario Fernandes      |           15 |
| Gustavo Borges..     |           16 |
| Valéria Santiago.... |           22 |
+----------------------+--------------+
6 rows in set (0.00 sec)
```

LTRIM, RTIM, TRIM

```
SELECT nome, LTRIM(nome), RTRIM(nome), TRIM(nome)
FROM usuario;
+----------------------+----------------------+--------------------+--------------------+
| nome                 | LTRIM(nome)          | RTRIM(nome)        | TRIM(nome)         |
+----------------------+----------------------+--------------------+--------------------+
| .Leonardo da Silva   | Leonardo da Silva    | .Leonardo da Silva | Leonardo da Silva  |
| . Cleber Duarte      | Cleber Duarte        | . Cleber Duarte    | Cleber Duarte      |
| Débora Rodrigues     | Débora Rodrigues     | Débora Rodrigues   | Débora Rodrigues   |
| Mario Fernandes      | Mario Fernandes      | Mario Fernandes    | Mario Fernandes    |
| Gustavo Borges..     | Gustavo Borges..     | Gustavo Borges     | Gustavo Borges     |
| Valéria Santiago. .. | Valéria Santiago.... | Valéria Santiago   | Valéria Santiago   |
+----------------------+----------------------+--------------------+--------------------+
6 rows in set (0.00 sec)
```

LTRIM, RTIM, TRIM

```
SELECT nome, LENGTH(nome), LENGTH(LTRIM(nome)), LENGTH(RTRIM(nome)), LENGTH(TRIM(nome))
FROM usuario;
```

nome	LENGTH(nome)	LENGTH(LTRIM(nome))	LENGTH(RTRIM(nome))	LENGTH(TRIM(nome))
.Leonardo da Silva	18	17	18	17
..Cleber Duarte	15	13	15	13
Débora Rodrigues	17	17	17	17
Mario Fernandes	15	15	15	15
Gustavo Borges..	16	16	14	14
Valéria Santiago.....	22	22	17	17

Fazer agora

Adaptar as seguintes alterações no script de criação do banco de dados utilizado no banco **cinema**:

1. Criar uma nova tabela **`usuario`**, contendo os campos **`id`**, **`nome`**, **`email`**, **`cidade_id`**.

2. Criar a tabela **`venda`** (com os campos **`data`**, **`hora`** e **`valorIngresso`**), relacionar esta tabela com a tabela **`sessao`**. Em seguida criar um relacionamento entre as tabelas **`usuario`** e **`venda`**.

3. Criar INSERT's de dados necessários a fim de cadastrar 10 usuários, e seguida cadastrar 50 vendas de ingressos;

1. Criar uma nova tabela **`usuario`**, contendo os campos **`id`**, **`nome`**, **`email`**, **`cidade_id`**.

```sql
-- usuario
DROP TABLE IF EXISTS usuario;
CREATE TABLE IF NOT EXISTS usuario (
  id INT NOT NULL AUTO_INCREMENT,
  cidade_id INT NOT NULL,
  nome VARCHAR(45) NULL,
  email VARCHAR(100) NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (cidade_id) REFERENCES cidade (id)
);
```

2. Criar a tabela **venda** (com os campos **data**, **hora** e **valorIngresso**), relacionar esta tabela com a tabela **sessao**. Em seguida criar um relacionamento entre as tabelas **usuario** e **venda**.

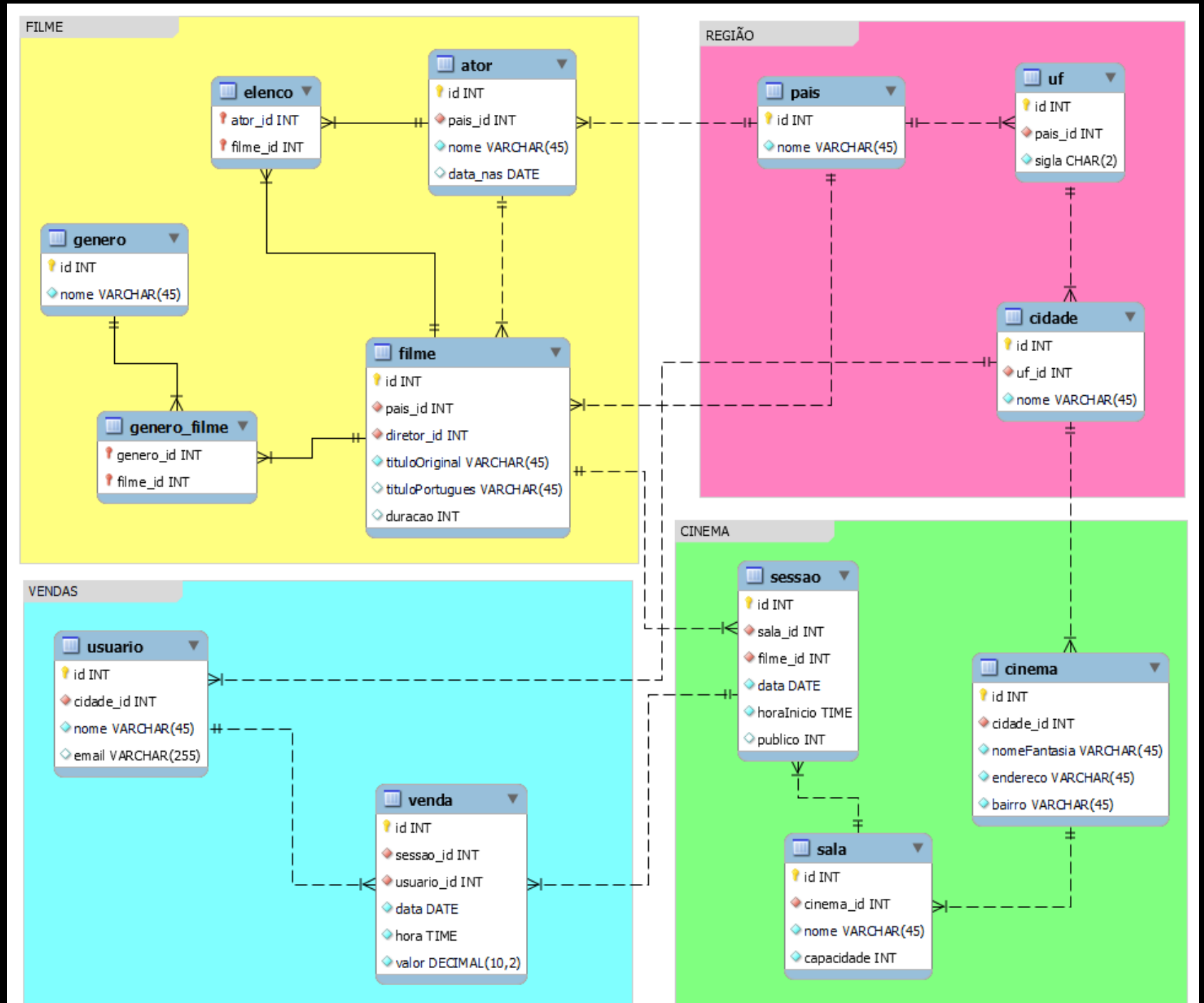
```sql
-- venda
DROP TABLE IF EXISTS venda;
CREATE TABLE IF NOT EXISTS venda (
  id INT NOT NULL AUTO_INCREMENT,
  sessao_id INT NOT NULL,
  usuario_id INT NOT NULL,
  data DATE NULL,
  hora TIME NULL,
  valorIngresso DECIMAL(10,2) NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (sessao_id) REFERENCES sessao (id),
  FOREIGN KEY (usuario_id) REFERENCES usuario (id)
);
```

3. Criar INSERT’s de dados necessários a fim de cadastrar **10 usuários**, e seguida cadastrar 50 vendas de ingressos

```sql
-- Usuário
INSERT INTO usuario(nome,email,cidade_id) VALUES('Leonardo Silva', 'leonardo@gmail.com',1);
INSERT INTO usuario(nome,email,cidade_id) VALUES('Cleber Duarte', 'cleber@gmail.com',1);
INSERT INTO usuario(nome,email,cidade_id) VALUES('Débora Rodrigues', 'debora@bol.com',2);
INSERT INTO usuario(nome,email,cidade_id) VALUES('Mario Fernandes', 'mario2@terra.com.br',2);
INSERT INTO usuario(nome,email,cidade_id) VALUES('Gustavo Borges', 'gugu@gmail.com',3);
INSERT INTO usuario(nome,email,cidade_id) VALUES('Valéria Santiago', 'val@gmail.com',3);
```

3. Criar INSERT’s de dados necessários a fim de cadastrar 10 usuários, e seguida cadastrar **50 vendas** de ingressos

```sql
-- Venda
INSERT INTO venda (sessao_id, usuario_id, data, hora, valorIngresso)
VALUES
(1,1,'2024-05-01','21:40:00',15.55),
(1,1,'2024-05-02','21:45:00',15.00),
(1,1,'2024-05-10','21:50:00',10.55),
(2,1,'2024-05-20','21:55:00',15.40),
(2,1,'2024-03-25','21:56:00',10.50),
(4,1,'2024-03-01','21:57:00',10.55),
(4,1,'2024-03-02','21:58:00',15.55),
(4,1,'2024-05-03','21:59:00',10.00),
(6,1,'2024-05-20','22:00:00',12.55),
(7,1,'2024-05-15','22:01:00',12.05),
(7,1,'2024-05-10','22:02:00',12.55),
(8,1,'2024-05-10','22:03:00',10.05),
(1,2,'2024-03-02','22:04:00',13.55),
(1,2,'2024-03-03','21:40:00',14.06),
(1,3,'2024-03-04','21:40:00',13.55),
(1,3,'2024-03-05','21:40:00',15.05),
(1,3,'2024-05-06','21:40:00',13.55),
...
(2,4,'2024-03-20','21:30:00',15.50);
```


FILME
elenco
ator_id INT
filme_id INT
ator
id INT
pais_id INT
nome VARCHAR(45)
data_nas DATE
genero
id INT
nome VARCHAR(45)
filme
id INT
pais_id INT
diretor_id INT
tituloOriginal VARCHAR(45)
tituloPortugues VARCHAR(45)
duracao INT
genero_filme
genero_id INT
filme_id INT
REGIÃO
pais
id INT
nome VARCHAR(45)
uf
id INT
pais_id INT
sigla CHAR(2)
cidade
id INT
uf_id INT
nome VARCHAR(45)
VENDAS
usuario
id INT
cidade_id INT
nome VARCHAR(45)
email VARCHAR(255)
venda
id INT
sessao_id INT
usuario_id INT
data DATE
hora TIME
valor DECIMAL(10,2)
CINEMA
sessao
id INT
sala_id INT
filme_id INT
data DATE
horaInicio TIME
publico INT
cinema
id INT
cidade_id INT
nomeFantasia VARCHAR(45)
endereco VARCHAR(45)
bairro VARCHAR(45)
sala
id INT
cinema_id INT
nome VARCHAR(45)
capacidade INT

Criar as seguintes consultas:

a. Crie uma consulta que liste todos os nomes dos filmes, gêneros e duração, ordenados por gênero e em seguida em ordem decrescente de tituloOriginal de Filme;

b. Selecionar os títulos dos filmes em ordem inversa de título;

c. Utilizando o operador IN, crie uma consulta para que liste o nome de todos os gêneros menos os gêneros de suspense, terror e comédia;

d. Crie uma consulta que retorne os títulos, gênero e duração de filmes em que o gênero seja DRAMA e a duração esteja entre 70 e 130 minutos;

e. Criar uma consulta para listar os títulos em original e português dos filmes que ainda não possuem ingressos vendidos;

f. Criar uma consulta para contabilizar o total dos valores de ingressos vendidos por filme, liste o nome do filme e a quantidade de ingressos;

g. Criar uma consulta que contabilize quantos ingressos foram vendidos por filme, liste o nome do filme e a quantidade, liste apenas os 5 primeiros caracteres do nome do filme e a quantidade de ingressos;

Criar as seguintes consultas:

h. Criar uma consulta que liste todos os nomes dos cinemas e a quantidade de caracteres de cada nome de cinema;

I. Modificar a consulta anterior para que liste apenas os 5 primeiros caracteres do nome do cinema listado;

j. Criar uma consulta para listar a quantidade de atores que trabalharam em cada filme. Listar o nome do filme e a quantidade de atores.

k. Crie uma consulta para atualizar o tempo para + 44 minutos em todos os filmes do gênero DRAMA.

l. Utilizando o IN, crie uma consulta que selecione o nome dos atores que não participaram de nenhum filme;

m. Utilizando o IN crie uma consulta que retorne o título e o gênero de todos os filmes que não passaram ainda em cinema algum.